

CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Previously Presented) A method comprising:
receiving source code;
transforming the source code to intermediate code;
executing the intermediate code based on external execution input;
generating data that indicates performance of the intermediate code when the intermediate code is executed with the external execution input; and
producing machine code based on the data and the intermediate code.
2. (Canceled).
3. (Previously Presented) The method of claim 1, wherein executing the intermediate code comprises simulating execution of the intermediate code.
4. (Previously Presented) The method of claim 1, wherein generating the data regarding the performance of the executed intermediate code comprises generating a performance profile.
5. (Previously Presented) The method of claim 4, wherein generating the data regarding the performance of the executed intermediate code further comprises annotating the intermediate code based, at least in part, on performance profile data.
6. (Original) The method of claim 5, wherein annotating the intermediate code comprises concatenating data structures that include the performance profile data to intermediate code to embed the performance profile data into the intermediate code.

7. (Original) The method of claim 5, wherein annotating the intermediate code comprises:
- generating a file that includes the performance profile data; and
 - mapping the performance profile data to corresponding portions of intermediate code.
8. (Previously Presented) The method of claim 5, wherein producing machine code based on the data and intermediate code includes providing the annotated intermediate code to a compiler, wherein the compiler produces the machine code based on annotated intermediate code.
9. (Original) The method of claim 5, wherein the performance profile data comprises one or more of branch statistics, loop statistics and function invocation statistics.
10. (Original) The method of claim 8, wherein the machine code executes faster than the intermediate code.
11. – 15. (Canceled).
16. (Previously Presented) The method of claim 1, further comprising:
- receiving the external execution input; and
 - using the external execution input to execute the intermediate code.
17. (Original) The method of claim 1, wherein the data comprises one or more of plain-text format, binary representations, database maps, and character delimited proprietary format.
18. (Previously Presented) A method comprising:
- transforming source code into intermediate code;

providing the intermediate code to a profiler that executes the intermediate code based on external execution input and generates annotated intermediate code based on the performance of the executed intermediate code when the intermediate code is executed with the external execution input;

receiving from the profiler the annotated intermediate code; and

transforming the annotated intermediate code into machine code.

19. (Original) The method of claim 18, wherein the annotated intermediate code is annotated to include one or more of branch statistics, loop statistics and function invocation statistics.

20. (Original) The method of claim 18, wherein providing the intermediate code to a profiler comprises providing the intermediate code to a virtual machine.

21. (Currently Amended) A method comprising:

producing intermediate code and machine code based upon source code;

receiving a data file generated by a profiler, wherein the data file indicates a

performance of the machine code as executed by the profiler;

producing modified intermediate code and modified machine code based on the source code and the data file; and

iteratively:

determining whether to produce further modified intermediate code and further modified machine code based upon whether a predetermined performance gain has been achieved in the modified machine code over the machine code; and, if the further modified intermediate code and the further modified machine code are to be produced:

providing the modified machine code to the profiler;

receiving another data file from the profiler; and

producing the further modified intermediate code and the further modified machine code based upon the source code and the another data file[.],

wherein the predetermined performance gain includes a percentage performance improvement based on execution speed.

22. (Original) The method of claim 21, further comprising providing the machine code to the profiler.
23. (Original) The method of claim 22, wherein providing the machine code to the profiler comprises providing the machine code to a virtual machine.
24. (Original) The method of claim 22, wherein providing the machine code to the profiler comprises providing the machine code to a probed processor.
25. (Original) The method of claim 21, wherein the data file includes one or more of branch statistics, loop statistics and function invocation statistics.
26. (Original) The method of claim 21, wherein the data file includes an identifier that associates an executed instruction with generated data.
27. (Canceled).
28. (Cancelled)
29. (Previously Presented) The method of claim 21, wherein determining whether the predetermined performance gain has been achieved comprises determining whether the modified machine code executes faster than the machine code.
30. (Previously Presented) The method of claim 21, wherein determining whether to produce further modified machine code comprises determining whether a cost of producing the further modified machine code exceeds a performance gain to be achieved by the modifying.

31. (Original) The method of claim 21, wherein receiving the data file comprises receiving the data file via one of a data storage device, an alphanumeric input device, a network interface, a shared data storage location, and a direct real-time connection.

32. – 40. (Cancelled)

41. (Previously Presented) An article of manufacture comprising:
a computer readable storage medium including thereon sequences of instructions that, when executed, cause an electronic system to:
receive source code;
produce intermediate code based on the source code;
execute the intermediate code based on external execution input;
generate performance data that indicates performance of the intermediate code when the intermediate code is executed with the external execution input; and
produce machine code based on the intermediate code and the performance data.

42. (Canceled).

43. (Previously Presented) The article of manufacture of claim 41, wherein the sequences of instructions that, when executed, cause the electronic system to generate the data regarding the performance of the executed code comprise sequences of instructions that, when executed, cause the electronic system to generate a performance profile.

44. (Original) The article of manufacture of claim 43, wherein the sequences of instructions that, when executed, cause the electronic system to cause the executed code to be modified based, at least in part, on the data comprise sequences of instructions that, when executed, cause the electronic system to annotate the intermediate code based, at least in part, on performance profile data.

45. (Previously Presented) The article of manufacture of claim 44, wherein the computer readable storage medium further comprises sequences of instructions that, when executed, cause the electronic system to provide the annotated intermediate code to a compiler, wherein the compiler transforms the annotated intermediate code into machine code.

46. (Currently Amended) An article of manufacture comprising:
a computer readable storage medium including thereon sequences of instructions that, when executed, cause an electronic system to:
produce intermediate and machine code based upon source code;
receive a data file generated by a profiler, wherein the data file indicates a performance of the machine code as executed by the profiler; and
produce modified intermediate code and modified machine code based on the source code and the data file; and
iteratively:
determine whether to produce further modified intermediate code and further modified machine code based upon whether a predetermined performance gain has been achieved in the modified machine code over the machine code; and, if the further modified intermediate code and the further modified machine code are to be produced:
provide the modified machine code to the profiler;
receive another data file from the profiler; and
produce the further modified intermediate code and the further modified machine code based on the source code and the another data file[.].
wherein the predetermined performance gain includes a percentage performance improvement based on execution speed.

47. (Previously Presented) The article of manufacture of claim 46, wherein the computer readable storage medium further comprises sequences of instructions that, when executed, cause the electronic system to provide the machine code to the profiler.

48. (Canceled).

49. (Original) The article of manufacture of claim 46, wherein receiving the data file comprises receiving the data file via one of a data storage device, an alphanumeric input device, a network interface, a shared data storage location, and a direct real-time connection

50. (Previously Presented) A system comprising:
a processor;
a dynamic random access memory coupled with the processor; and
an article of manufacture comprising a computer readable storage medium including thereon sequences of instructions that, when executed, cause an electronic system to:
receive source code;
produce intermediate code based on the source code;
execute the intermediate code based on external execution input;
generate data that indicates performance of the intermediate code when the intermediate code is executed with the external execution input; and
produce machine code based on the data and the intermediate code.

51. – 52. (Canceled).

53. (Previously Presented) The system of claim 50, wherein the computer readable storage medium further comprises sequences of instructions that, when executed, cause the electronic system to:
receive external execution input; and
use the external execution input to execute the intermediate code.